

Travail 3

Partie 2

Pondération: 16% au total (50% partie 1 et 50% partie 2)

Remise : jeudi 25 avril avant minuit (via Léa)

Enlevez le .db, le ipch et les dossiers debug

Exigences

- Chaque classe doit être programmée à l'intérieur de ses propres fichiers (.h et .cpp)
- Vous devez faire des classes C++ standard.
- **Utilisez vos objets le plus souvent possible et non les valeurs affichées à l'écran.**
- Respectez les standards de programmation et les bonnes techniques de programmation.
- Respectez les principes de la programmation objet.

Description de la partie 2

Dans la partie 1 du travail nous avons programmé différents types de véhicules parmi lesquels il y a des véhicules d'urgence.

Nous allons maintenant ajouter différents types d'urgence dans le travail (voir schéma en annexe). Le bon type de véhicule devra alors se déplacer vers l'urgence en question afin de la traiter.

Programmation des classes

- Programmez les classes respectant le schéma présenté en annexe.
- Faites les constructeurs par défaut, avec paramètres et par copie dans chaque classe de la hiérarchie des urgences. N'oubliez pas le destructeur.
- Ajoutez des méthodes dans la hiérarchie des véhicules afin de :
 - Déterminer si un véhicule peut répondre à un type d'urgence donné. La méthode doit donc recevoir l'urgence en paramètre afin de déterminer si elle est en mesure de la traiter.
 - Les ambulances peuvent répondre aux urgences médicales.
 - Les pompiers peuvent répondre aux urgences de type feu.
 - Aucun véhicule ne peut répondre aux urgences nationales.
 - Même si ce n'est pas le cas actuellement, il n'est pas exclu qu'un type de véhicule puisse répondre à plus d'un type d'urgence.
 - Dans le cas où votre nouveau type de véhicule est un véhicule d'urgence alors celui-ci ne peut traiter aucun type d'urgence.
 - Calculer le coût qu'engendrera le traitement de l'urgence par le véhicule en question.
 - Pour calculer le coût, nous avons besoin du coût de base de l'urgence ainsi que de la distance à parcourir pour se rendre à l'urgence en question. La méthode pour calculer le coût doit donc recevoir l'urgence à traiter, laquelle contient toutes les valeurs nécessaires.
 - Pour calculer la distance entre le véhicule et l'urgence à traiter, utilisez l'équation mathématique permettant de calculer la distance entre 2 points :
$$\sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$$
 - Vous pouvez utiliser les fonctions pow() et sqrt() de la librairie math.h

- Une ambulance engendre des coûts équivalents à : coût de base * distance
- Un camion de pompier (plus gros, consomme plus) engendre des coûts équivalents à : coût de base * distance * 2
- Dans Donnees :
 - Déclarez un tableau permettant de garder les urgences.
 - Modifiez le constructeur et le destructeur de Donnees afin de tenir compte du nouveau tableau.
 - Ajoutez une méthode InitialiserLesUrgences().
 - La méthode doit ajouter des urgences à l'aide du constructeur avec paramètres. La méthode doit ajouter au moins 4 urgences médicales, 4 urgences de type feu et une urgence nationale.
 - Appelez InitialiserLesUrgences() à partir du constructeur de Donnees.

Fonctionnalités à ajouter

- Dans la Form, ajoutez une liste déroulante contenant le numéro de toutes les urgences.
- Vous pouvez ajouter une variable globale servant à garder l'urgence courante.
- Notez que les urgences ne sont pas représentées sur la Form.
- Lorsqu'une urgence est choisie dans la liste, l'immatriculation des véhicules en mesure de répondre à cette urgence doivent apparaître dans une 2^{ème} liste déroulante.
- Lorsqu'un véhicule est choisi dans cette 2^{ème} liste :
 - Le coût pour répondre à l'urgence doit s'afficher dans une zone de texte.
 - Le véhicule choisi doit se déplacer à l'endroit où l'urgence se déroule.
 - L'urgence doit disparaître du tableau des urgences et de la liste déroulante.

Critères de correction de cette partie :

- Respect du schéma de classes
- Application des règles de la programmation objet et respect des pratiques en orienté objet
- Respect du standard de programmation et des bonnes pratiques de programmation
- Utilisation adéquate des classes et des objets
- Fonctionnement selon les règles énoncées et respect des exigences demandées
- Répartition des points de cette partie :
 - Programmation des classes de la hiérarchie Urgence **20 pts**
 - Création des urgences dans Donnees **15 pts**
 - Trouver les véhicules pouvant répondre à l'urgence..... **30 pts**
 - Calcul du coût de l'urgence..... **20 pts**
 - Déplacement du véhicule et destruction de l'urgence **15 pts**

Schéma des classes partie 2

